Data Exploration Analysis on Store Sale Data

San Jose State University

Le Dao (le.h.dao@sjsu.edu)
Dang Minh Nhu Nguyen (dang.m.nguyen@sjsu.edu)
Johanna Chen (johanna.chen@sjsu.edu)
Andrew Pickard-Christen (andrew.pickard-christen@sjsu.edu)

MATH 167PS SEC 01

Dr.Bremer Martina

May 11, 2022

Table of Contents

I. Introduction:

Holidays are often busy snippets of the year where people come together and celebrate the special day. For instance, many holidays throughout the globe are filled with banquets of food and gifts. Because there seems to be such a drastic increase in consumer demands during the holidays, we wondered if holidays do impact consumer behavior and even oil prices. To explore this question, our group decided to work with a dataset from a competition on Kagle called 'Store Sales-Time Series Forecasting'. The data contains information about store transactions, sales in stores, holidays, and oil prices in Ecuador. Ecuador is a culturally rich country that is located in South America. Ecuadorians usually commemorate holidays with music and dance festivals [3]. Because holidays are accompanied by major events such as parades, typically store owners close their business for the time being and join in on the party [3]. Even basic service companies like banks or government systems like public transportation are often closed. At the heart of Ecuador is Quito, the capital that has the leading population count. As we later on see, most of the stores in our dataset are based in Quito. Our group chose to work with the following four csv files: holidays_events, stores, transactions, and oil. The time frame of the data spreads across four full years, ranging from 2013 to 2017. This allowed our group to observe the annual changes across different variables. As we became more familiar with the data, we came up with a few questions that will be analyzed and answered with visual aid in our paper. The questions we will answer are the following: How does oil price change in different locations? Which type of store has the most transactions during the holiday season? Is consumer behavior generally the same during holidays and non-holidays? Do holiday events affect store transactions? If so, how? We are interested in answering these questions to better understand and support the economy. By knowing how the country is financially doing, we can more accurately predict sales in the future to meet consumer demands. This prediction accuracy may also increase profit because it avoids overestimating demand, which may result in a wasteful surge of supply.

II. Materials and Methodology:

1. Materials and Data Cleaning:

"Pandas", "fancyImpute", "matplotlib", "plotly" and "seaborn" were used for this project. Pandas was imported to assist the process of manipulating and assessing the datasets. It allows users to perform operations faster and gives them more efficient and flexible methods to subset and merge the data [4]. The fancyImpute package was imported so that we may use chained equations to perform multiple imputations, sometimes known as mice. Matplotlib, plotly, and seaborn were imported for the goal of providing meaningful plots.

Kaggle, a popular forum for Data Scientists and Statisticians to share ideas and practice, provided the data [7]. As mentioned above, for this project, we planned to use 4 datasets (i.e. transaction.csv, holiday_events.csv, oil.csv, stores.csv) from a Store Sale data competition held by Kaggle. To reach the goal, we must first comprehend the data. To begin, ensure that each feature in each dataset has the correct format after being put into the Jupyter Notebook. If the merging columns did not have the same data type, an error might be signaled. From figure 1, it was clear that they were all correctly stored, and no further reformatting was required. The next

step was to discover the common factors that would help us connect all four datasets. Indeed, the date variable was shared by holiday events, oil, and the transaction dataset; the transaction dataset also had store number as a connection attribute to stores.csv. The data was merging specifically on the left. If this was not specified, the final dataset would include around 13000 observations. Meanwhile, if the technique was specified, roughly 85000 observations will be exported, preventing data loss. Further, because certain columns had ambiguous names, it was critical to change the names of the features to help readers understand what the columns represented. Column “store_nbr”, “type_y”, “type_x”, “dcoilwtico” were renamed to “Store_number”, “Holiday_type”, “Store_type” and “Oil_price” respectively.

```
Store_number        0
City                0
State               0
Store_type          0
Cluster             0
Date                0
Transactions        0
Holiday_type    71096
Locale          71096
Locale_name     71096
Description     71096
Transferred     71096
Oil_price       26422
dtype: int64
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54 entries, 0 to 53
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   store_nbr  54 non-null     int64
 1   city       54 non-null     object
 2   state      54 non-null     object
 3   type       54 non-null     object
 4   cluster    54 non-null     int64
dtypes: int64(2), object(3)
memory usage: 2.2+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 350 entries, 0 to 349
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   date         350 non-null    object
 1   type         350 non-null    object
 2   locale       350 non-null    object
 3   locale_name  350 non-null    object
 4   description  350 non-null    object
 5   transferred  350 non-null    bool
dtypes: bool(1), object(5)
memory usage: 14.1+ KB
```

Figure 1: Total missing value after merging, store.info(), holiday.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 83488 entries, 0 to 83487
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   date          83488 non-null  object
 1   store_nbr     83488 non-null  int64
 2   transactions  83488 non-null  int64
dtypes: int64(2), object(1)
memory usage: 1.9+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1218 entries, 0 to 1217
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   date        1218 non-null   object
 1   dcoilwtico  1175 non-null   float64
dtypes: float64(1), object(1)
memory usage: 19.2+ KB
```

transaction.info(), oil.info()

New columns called day, year, and month had been added using the command pd.DatetimeIndex in order to provide a comprehensible graph and an efficient way to access the data.

From figure 1, because no record was available for those days, a huge number of NaN values were detected. We would develop biases and reduced the accuracy of further analysis if we ignored all the rows with missing data. Realizing the need for Data Cleaning, we then processed to perform imputation on columns that had the same missing values (i.e. Holiday_type, Locale, Locale_name, Description, and Transferred). Since those missing attributes were not holidays, we decided to run a for loop to fill in all NaN with ‘No Holiday/Event’. For the Oil_price column, we took an extra step of data imputation using package “fancyimpute” and function “MICE().fit_transform” to compute the mean values within each month of the year and appended those mean to the missing price accordingly to month and year.

The holiday_type contains the values that are quite abstract to comprehend; hence, further interpretation on these columns was conducted. In holidays_type, there were six types of day:

Work Day, Holiday, Transfer, Bridge, and Additional. Transferred days were holidays that appeared on the respective calendar day, but were moved by the government to another day. “Transfer” day types were the dates that transferred holidays were actually celebrated. “Bridge” days were additional days that prolonged the holiday season, like a break extension. To make up for these missing work days, “Work Day” day types were scheduled days for employees to work when they typically do not. For instance, if Christmas was on a Wednesday and break was prolonged till the weekend, Thursday and Friday would be considered “Bridge” days and the weekend would be “Work Day” days where people usually do not work. “Additional” days are extra days attached to the holidays on the calendar, for example, Christmas Eve to Christmas. Lastly, we added our own day type “No Holiday/Event”.

The figure below shown the data frame that would be used for plotting purposes.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85007 entries, 0 to 85006
Data columns (total 17 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    85007 non-null  int64
 1   Store_number  85007 non-null  int64
 2   City          85007 non-null  object
 3   State         85007 non-null  object
 4   Store_type    85007 non-null  object
 5   Cluster       85007 non-null  int64
 6   Date          85007 non-null  object
 7   Transactions  85007 non-null  int64
 8   Holiday_type  85007 non-null  object
 9   Locale        85007 non-null  object
 10  Locale_name   85007 non-null  object
 11  Description   85007 non-null  object
 12  Transferred   85007 non-null  object
 13  Oil_price     85007 non-null  float64
 14  year          85007 non-null  int64
 15  month         85007 non-null  int64
 16  day           85007 non-null  int64
dtypes: float64(1), int64(7), object(9)
memory usage: 11.0+ MB
```

Figure 2: Final Dataset

2. Methodology and Interpretation:

We were interested in comparing consumer behavior during the holiday season and normal working days. To do this, we compared the total transactions by day type and year. Because there were so many day types, our group had to specify which days we considered holidays. While merging the data, we realized that merging on date with the holidays_events file loses a lot of information. The holidays_events file was mainly used to gather information on holidays, and thus combining on date with other datasets eliminates many other regular working days. As a result, we left-merged the datasets together to maintain the data, adding “No Holiday/Event” to all the missing values in the day type column. Our group agreed to define holiday as all the above except for “Holiday/Event” and “Work Day”.

From the main data after cleaning, I chose to work with columns type_y, year, transactions, and date. First, I identified the unique years in the "year" column. Then I masked the data to contain only non-holidays for each specific year and filtered it to compute the sum of all the transactions using both the "type_y" and "transactions" column. The same was done for holidays. Both the yearly total transactions for holidays and non-holidays were plotted side-by-side to compare the behavior of consumers between these two day types. The bar plot is displayed below. As we see, consumers generally make less total transactions during holidays than non-holidays. The graph suggests approximately a 1 to 5 ratio of holiday transactions versus non-holiday transactions. Because of the limited data we have, not every day of the year has recorded transactions. Thus, the graph below may be deceptive. To consider the data we are working with, I filtered out the specific dates that are categorized as holidays. I then extracted the specific dates, excluding year, over all data considered holidays. I carried out the same process with non-holiday days. In total, the dataset had 105 distinct holiday dates recorded and 328 distinct non-holidays dates recorded. This tells us that if consumers generally behaved the same way during the holidays as normal working days, we would more likely see a 1 to 3 ratio. However, our 1 to 5 ratio illustrates that there are less transactions being made on holidays than the average working day. This may be due to personal reasons like staying at home during the holidays. On a societal level, stores may be closed on the holidays and thus consumers who may want to buy things from certain stores cannot.

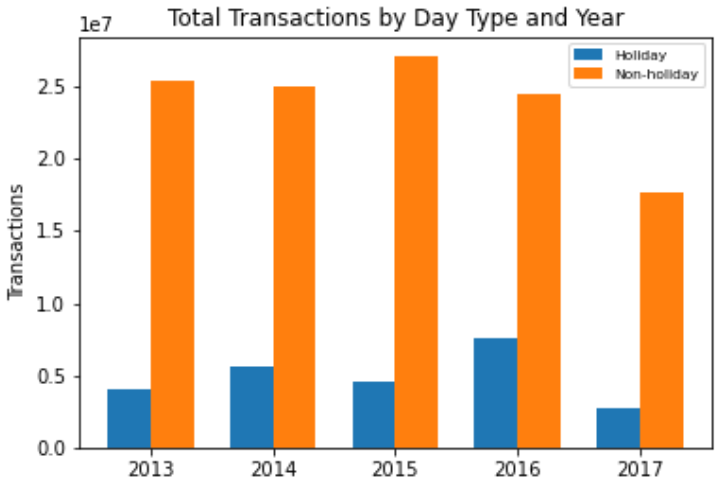

Figure 3: Total Transactions by Day Type and Year

After having in-depth observations on different types of food and the stores that sell them, the next step was comprehending how the average transactions altered on certain dates in different cities in Ecuador. To be able to see the average number of transactions per holiday type per city, using Python to create another dataframe that groups those 2 columns. Since the resulting graph needed to illustrate all 2 categorical variables, which was holiday and city, along with 1 numeric variable, transactions, a scatter plot generated by plotly package would be able to convey all the necessary characteristics.

Based on figure 4, Cayambe and Quito were the two cities that had the highest average transaction since they were both Ecuador's biggest tourist attractions[1]. Many visitors came to these places for traveling, hence, even when no holiday/event took place, their average transaction was approximately double the amount of transactions in other cities. In other cities, the overall trend was that the people seemed to purchase more before the holiday/event took place (additional category has highest transaction) or few days after the holiday occurred (number of transactions in Bridge came in second place). While on no holiday/event day, the average transactions dropped drastically.

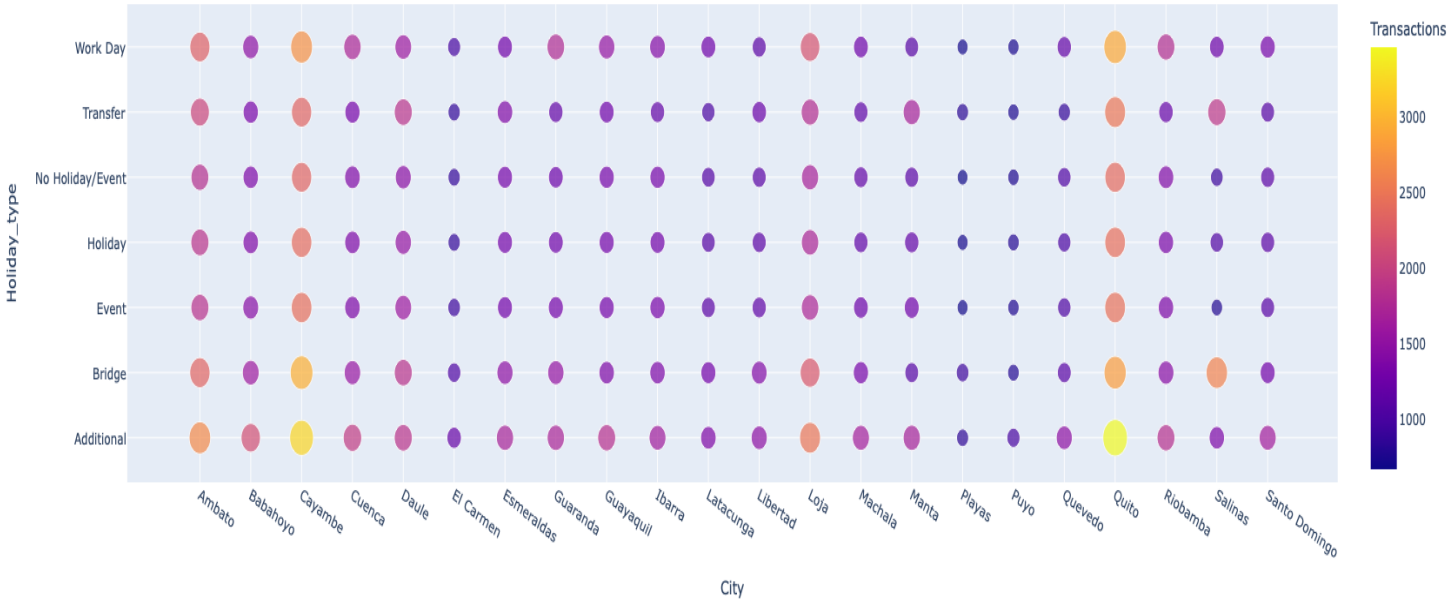

Figure 4: Averages Transactions of City and Holiday Type [2]

Figure 5 shows the number of stores that each city in Ecuador had. The datasets recorded the transactions of 54 stores, yet, 17 of them were located in Quito, which accounted for half of the total stores of this dataset. For the 21 remaining cities, each of them only had 1-2 stores being recorded. Knowing that Quito was the capital, however, such a big gap in the number of stores being recorded between Quito and other cities might lead to a biased dataset.

Another interesting feature from figure 4 and figure 5 is that while Cayambe only had 2 stores and Guayaquil had 7 stores, the average transactions in Cayambe were much higher than that of Guayaquil. This might be because Cayambe was a bigger tourist attraction compared to Guayaquil [1]. The more travellers there were, the greater the number of transactions.

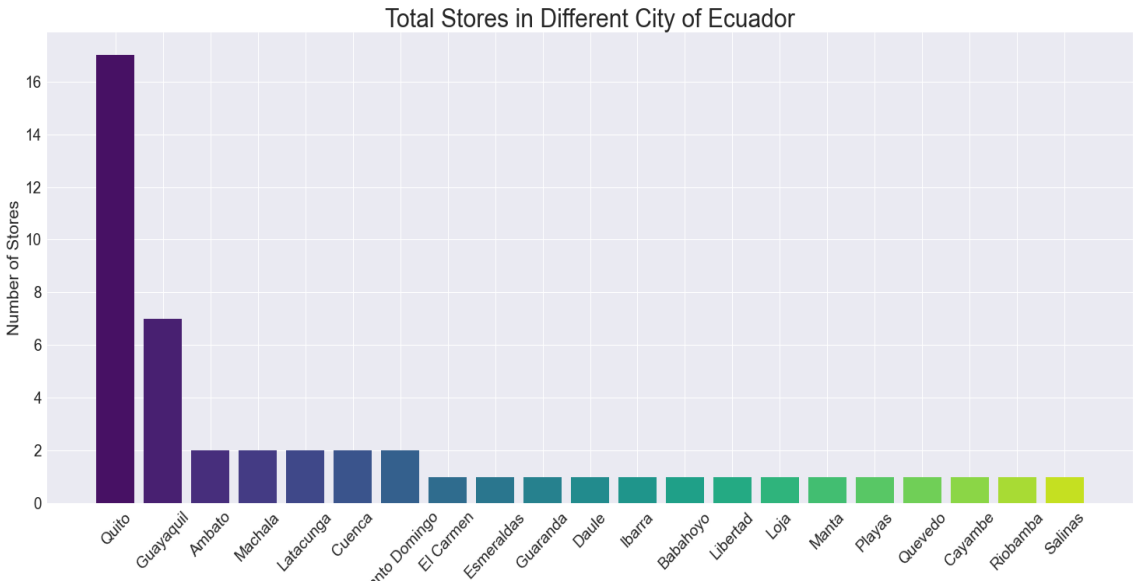

Figure 5: Number of stores in Each city

In addition to observing the transactions based on location and holiday, seeing how it changed throughout every month from 2013 to 2017 might provide some insights on the consumers' behavior of Ecuador people. As we wanted to plot the transactions of all 5 years for all 12 months, columns month and year were merged and aggregated the average transaction. A line plot is the best way to illustrate this issue.

From 2013 to 2016, the average transactions experienced a significant decline. This happens as in 2016, a disastrous earthquake, 7.8-magnitude, occurs and takes away the lives of hundreds of people living in Ecuador. However, the country's economy is able to revive within the next year as the number of transactions seems to go up again in 2017 [5].

When looking at the price every month, they seem to have a common trend. The number of transactions is not so high, yet, it is not the lowest in January. It slightly increases as time moves forward to April and May, and then plummets in the next few months. Not until between November to December, the time where Christmas and New Year's Eve occur, did the transactions' volume hit its peak.

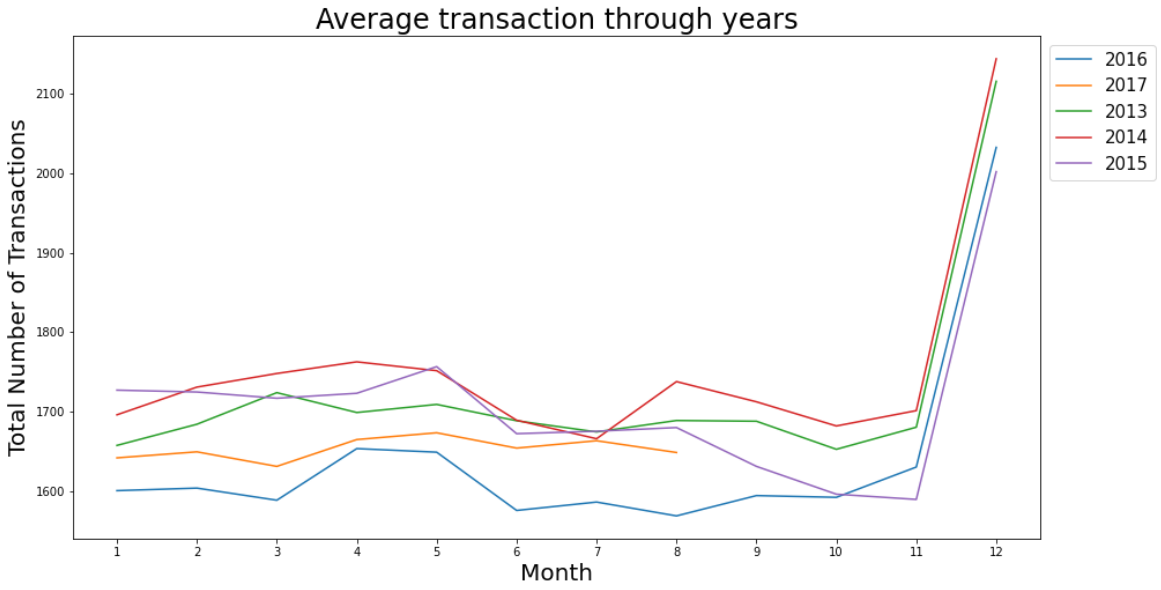

Figure 6: The average transactions throughout the year

The next goal is to explore the data to see if there is any effect of holiday on each cluster group, where each cluster was identified by what type of products they were offering (i.e. Grocery, Clothing, Auto). After obtaining the cleaned data from the previous steps, we were able to group them by cluster and store to explore which stores belong to which cluster; due to the fact that we were not planning to use the "train.csv" and "test.csv" datasets, based on Kaggle description, we would identify some of the products that each cluster was currently holding, the detailed result are summarized in Table 1.

Table 1: Summary Cluster Group and Store Number [5]

Cluster Group	Store Number	Type of Product
1	24, 25, 27	Beverages, bakery, books,...
2	37, 42	Dairy, ladies wear, meats,...
3	16, 30, 32, 33, 35, 40, 54	Books, celebration, cleaning, frozen food,…
4	5, 38, 41	Personal care, liquor, wine, beer,...
5	44	Meats, personal care, home care,...
6	9, 11, 20, 21, 34, 39	Beauty, beverages, automotive, baby care,…
7	14, 22	Beauty, school and office supplies, poultry,...
8	3, 7, 8	Magazines, players and electronics, hardware,…
9	4, 23	Lawn and garden, magazines, books, cleaning,…
10	26, 28, 29, 31, 36, 43	Deli, bakery, home appliances,...
11	45, 49, 52	Hardware, frozen foods, lingerie,...
12	17	Personal care, pet supplies, ladies wear…
13	1, 2, 6, 53	Automotive, baby care, beauty, dairy, hardware,…

14	46, 47, 48, 50	Celebration, bakery, pet supplies,...
15	10, 12, 13, 15, 19	Prepared food, seafood, school and office supplies,…
16	18	Automotive, prepared food, produce,...
17	51	Baby care, automotive, poultry,...

Since we were interested in the total transactions which were separated by type of holidays (i.e. Local, National, Regional), the choice of stacked bar plot would be meaningful in this case. By subsetting the cleaned data based on holiday types and using that subset data to visualize the total transactions between each cluster group (i.e 1,2,3…17), we were able to obtain the result plot *Figure 7*.

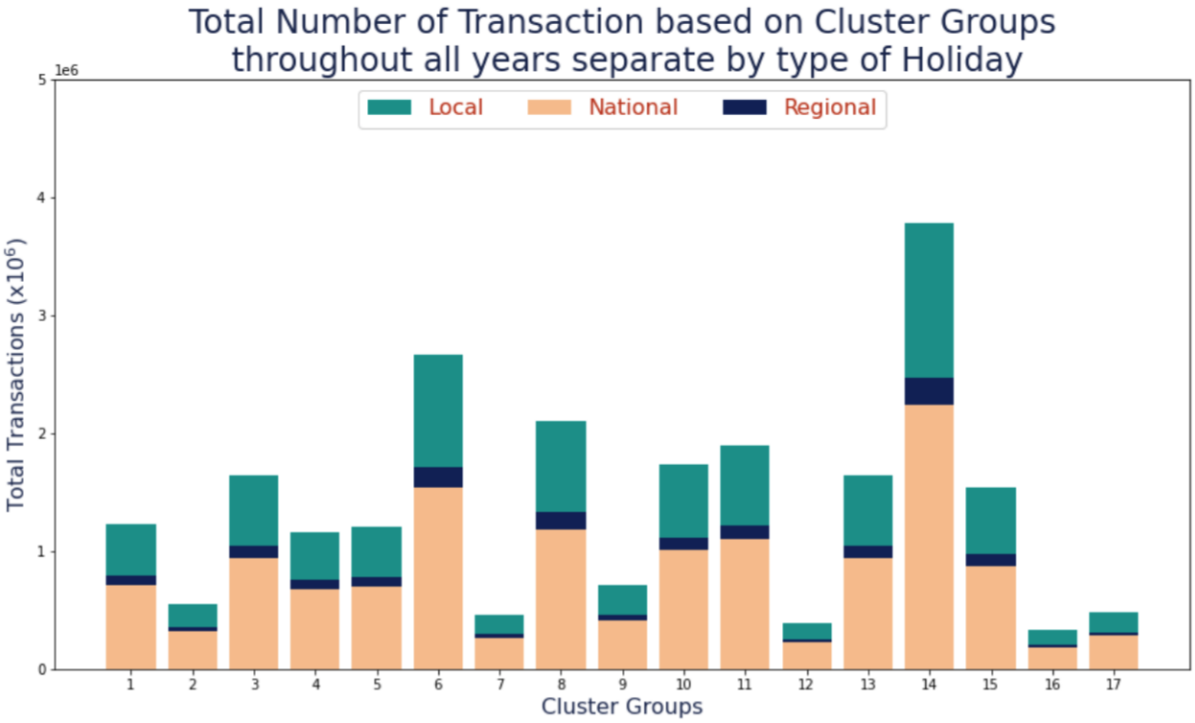

Figure 7: Total number of Transactions based on Cluster Groups throughout all years

From the plot, cluster group 14 (i.e. store 46, 47, 48, 50) was the one who had the highest total transaction in all 3 types of holidays as well as in each individual holiday; followed by group 6 and 8. Meanwhile, cluster group 12 (i.e. store 17), group 16 (i.e store 18), group 17 (i.e store 51) have the lowest total transactions recorded. It is because these clusters had only 1 store each, its results seem reasonably accurate. Then, we decided to have a deeper check in which city the store was located, in which holidays the locals would like to purchase more, and how much the oil price was during that time. Thanks to the help of Python functions (i.e. max, sort value,...), it was easy for us to output those values in *Figure 8*.

Store_number	Locale	City	State	Date	Transactions	Locale_name	Description	Oil_price
46	Regional	Quito	Pichincha	2017-06-25	5244	Santo Domingo de los Tsachilas	Provincializacion de Santo Domingo	107.04
	No Holiday/Event	Quito	Pichincha	2017-08-14	6506	No Holiday/Event	No Holiday/Event	110.62
	Local	Quito	Pichincha	2017-08-15	7008	Santo Domingo	Traslado Fundacion de Guayaquil	107.43
	National	Quito	Pichincha	2017-08-11	8001	Ecuador	Viernes Santo	107.95
47	Regional	Quito	Pichincha	2017-06-25	5247	Santo Domingo de los Tsachilas	Provincializacion de Santo Domingo	107.04
	No Holiday/Event	Quito	Pichincha	2017-08-14	6130	No Holiday/Event	No Holiday/Event	110.62
	Local	Quito	Pichincha	2017-08-15	6791	Santo Domingo	Traslado Fundacion de Guayaquil	107.43
	National	Quito	Pichincha	2017-08-11	7727	Ecuador	Viernes Santo	107.95
48	Regional	Quito	Pichincha	2017-06-25	4541	Santo Domingo de los Tsachilas	Provincializacion de Santo Domingo	107.04
	No Holiday/Event	Quito	Pichincha	2017-08-14	5830	No Holiday/Event	No Holiday/Event	110.62
	Local	Quito	Pichincha	2017-08-15	6264	Santo Domingo	Traslado Fundacion de Guayaquil	107.43
	National	Quito	Pichincha	2017-08-11	7044	Ecuador	Viernes Santo	107.95
50	Regional	Ambato	Tungurahua	2017-06-25	3868	Santo Domingo de los Tsachilas	Provincializacion de Santo Domingo	107.04
	No Holiday/Event	Ambato	Tungurahua	2017-08-14	4241	No Holiday/Event	No Holiday/Event	110.62
	Local	Ambato	Tungurahua	2017-08-15	4585	Santo Domingo	Traslado Fundacion de Guayaquil	107.43
	National	Ambato	Tungurahua	2017-08-11	5456	Ecuador	Viernes Santo	107.95

Figure 8: Cluster 14's Store, Holiday and Oil Price information

Interestingly, Quito is where stores 46, 47, and 48 were located, which also from the previous analysis, we learned that Quito has the record of highest transaction city. People tends to spend the most during Provincializacion de Santo Domingo (i.e. Regional Holiday) which occurred on 06/25/2017, Viernes Santo (i.e. National Holiday) which were on 08/11/2017, and for Local Holiday named Traslado Fundacion de Guayaquil which were on 08/15/2017 people tend to prepare on the same day and also the previous day (e.g. 08/14/2017). Even though the oil price was high, approximately 110 unit currency, people were still willing to purchase a lot in order to celebrate the holidays. This in fact, enhances the proof of uncorrelated between the oil price and transaction. Another interesting fact worth mentioning is 08/2017 is not only the highest number of transactions date record but was also the one of the last day recorded.

The final goal of the data was to observe the change in oil prices over multiple years, shown by the box plot below. The data was aggregated over every day of the year and over every store, as to give the most accurate representation of the data.

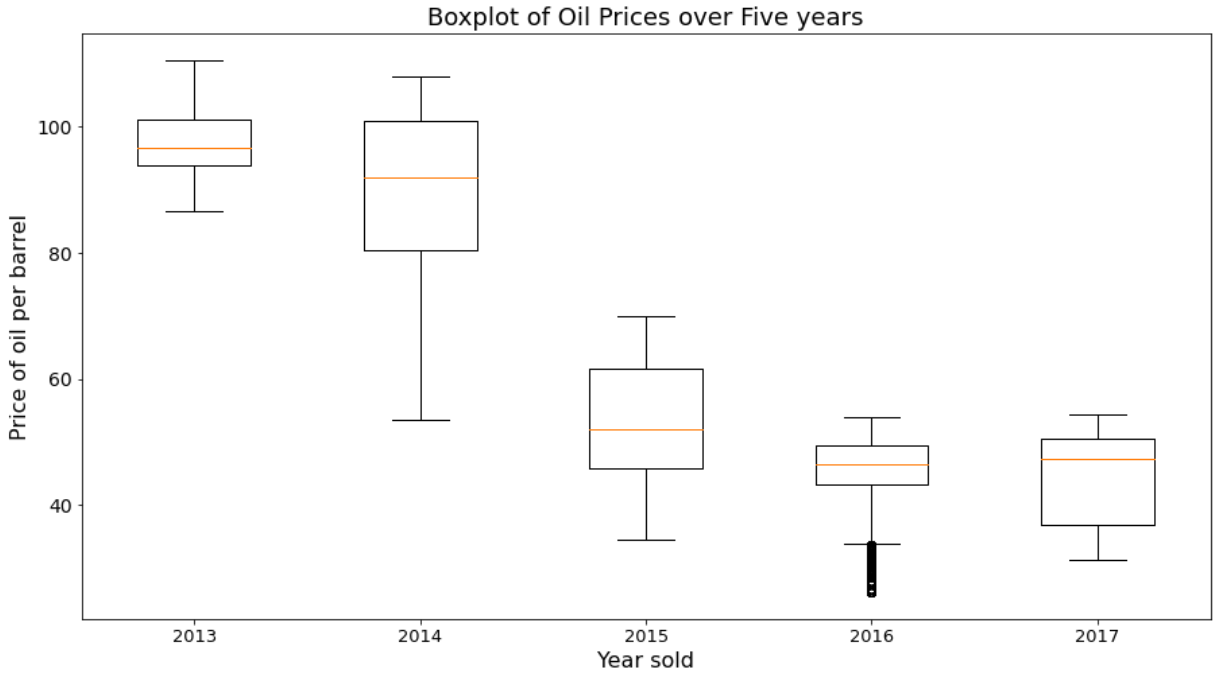

Figure 9: Changing prices of crude oil across the five observed years

Figure 8 shows that the price went through a severe decline from 2014 to 2015, known as a commodity price shock, or simply a time when the prices for commodities have radically increased or decreased in a short amount of time. This particular commodity shock was due to the oversupply of crude oil with respect to its demand. The figure above shows that the price unfortunately failed to recover by even three years later after its record high of well over one hundred dollars in 2013. Due to limitations of the data recorded, 2017 only contains a portion of the true daily oil price, so it is unknown whether or not the price recovered later that year, however unlikely. However, according to oilprices.com, the price of WTI crude, the specific producer of oil used in the dataset, is at $102 per barrel, so the price has successfully bounced back since this point.

III. Conclusion:

To summarize, our data was instrumental in discovering the trends taken by prices of and transactions involving crude oil in Ecuador. The trends observed are consistent with the holidays and customs of the Ecuadorian people, and those monthly breaks are reflected in the data we collected and tidied. The conclusions we were able to draw would be able to be used for a multitude of other purposes using computing languages like python and associated graphing packages. The results we compiled from researching the data can be used to predict other aspects of oil inside, and perhaps outside, the country of interest.

IV. References

[1] Dearsley, Bryan. (2021, Sept. 24). *10 Top-Rated Tourist Attractions in Ecuador: Planetware.* PlanetWare. https://www.planetware.com/tourist-attractions/ecuador-ecu.htm

[2] Discrete. Discrete colors in Python. (n.d.). Retrieved May 11, 2022, from https://plotly.com/python/discrete-color/

[3] *Holidays and Festivals in Ecuador*. ANYWHERE ECUADOR. https://www.anywhere.com/ecuador/travel-guide/holidays-and-festivals.

[4] Johnson, Daniel. (2022, March 8). *Python Pandas Tutorial: DataFrame, Data Range, Use of Pandas.* Guru99. https://www.guru99.com/python-pandas-tutorial.html#:~:text=In%20a%20nutshell%2C%20Pandas%20is,perform%20operations%20on%20these%20structures

[5] Reid, Kathryn. (2018, July 9). *2016 Ecuador earthquake: Facts, FAQs, and how to help*. World Vision. https://www.worldvision.org/disaster-relief-news-stories/2016-ecuador-earthquake-facts

[6] *Store Sales - Time Series Forecasting.* Kaggle. https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data?select=oil.csv

[7] Moyer, E. (2017, March 8). *Google buys Kaggle and its gaggle of AI geeks*. CNET. https://www.cnet.com/science/google-buys-kaggle-and-its-gaggle-of-ai-geeks/